COBB
INSIDE DOS®™

October 1994
Vol. 5 No. 10

Tips & techniques for MS-DOS & PC-DOS Versions 5 & 6

DOS 6 helps those who HELP themselves

VERSION
6.x

Have you ever wanted to look up the syntax for a DOS command but realized that your DOS manual wasn't handy? If you're an experienced DOS user, you probably turned to DOS's online help system to get the information you needed.

However, if you're new to DOS, you may not realize how much information is literally at your fingertips. In this article, we'll show you how to use the HELP command to obtain a quick onscreen explanation of nearly every DOS command and device driver.

The DOS 5 and DOS 6 difference

As many of you know, Microsoft introduced DOS's first online help system in version 5. This help system lets you quickly view the syntax of any DOS command when you enter *help* followed by the name of the command you need help with, or when you enter the name of the command you need help with followed by a slash and a question mark. (See "Getting Help at the Command Prompt" in the July 1991 issue of *Inside DOS*.)

DOS 6's online help system elaborates on its younger sibling—courtesy of an extensive hypertext help system. Now, besides giving you loads of information about nearly every command and device driver in DOS 6, the latest help system lets you perform searches on the help text in order to locate a particular string.

In addition, Microsoft developers added four switches to the HELP command to make viewing the help text a little easier. These switches, which we'll discuss later in this article, let you customize the way your screen displays the HELP command's output.

HELPing yourself

As with DOS 5, you can get onscreen information about a command by issuing the HELP command at the DOS 6 command prompt in the form

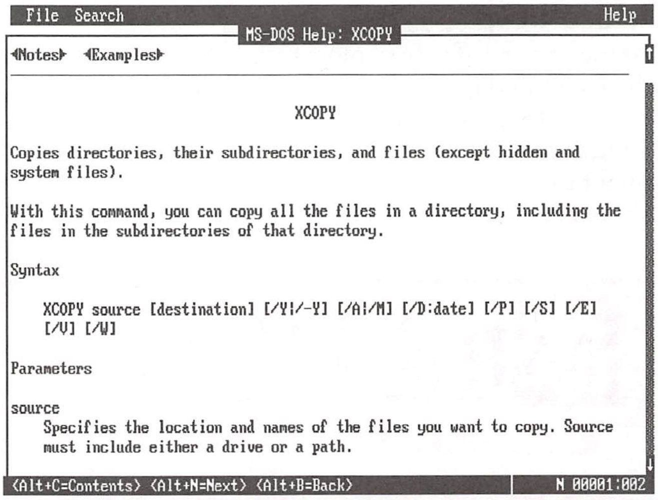
```
help command
```

where *command* is the name of the command with which you need help. For example, if you're unsure about the correct syntax of the XCOPY command, you can enter

```
c:\>help xcopy
```

at the command prompt. When you do, DOS will open the Help screen shown in Figure A.

Figure A



```
                                 XCOPY

Copies directories, their subdirectories, and files (except hidden and
system files).

With this command, you can copy all the files in a directory, including the
files in the subdirectories of that directory.

Syntax

    XCOPY source [destination] [/Y|/-Y] [/A|/M] [/D:date] [/P] [/S] [/E]
    [/V] [/W]

Parameters

source
    Specifies the location and names of the files you want to copy. Source
    must include either a drive or a path.
```

DOS 6 offers online help for nearly every command and device driver.

IN THIS ISSUE

A Publication of The Cobb Group

When you enter a DOS command followed by a slash and a question mark, DOS 6 will execute the FASTHELP command. FASTHELP displays an abbreviated listing of a command's syntax and switches. (You can also execute the FASTHELP command by entering *FASTHELP* and the name of the command you want help with. See "Using FASTHELP to View Short Help Messages" in the March 1994 issue of *Inside DOS*.)

For example, if you want to view the help text that FASTHELP displays for the XCOPY command, you enter the command

```
c:\>xcopy/?
```

or

```
c:\>fasthelp xcopy
```

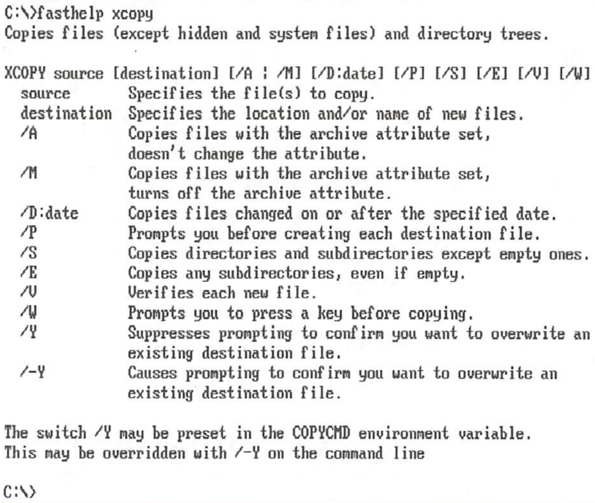
DOS will display the list shown in Figure B.

More help

At the upper-left side of most Help screens, you'll see the Notes and Examples options. When you select the Notes option, DOS will display help text that goes beyond explaining the command's syntax. And as you might expect, when you select Examples, DOS will show you examples of how you can apply the command and its switches. (If there's only one note or example, DOS will display the option Note or Example.)

To demonstrate, type *help xcopy*. When you do, DOS will display the Help screen containing information on the syntax of the XCOPY command, as shown in Figure A. Next, to see how to use the XCOPY command, select Notes (or press N, then [Enter]) at the

Figure B



```
C:\>fasthelp xcopy
Copies files (except hidden and system files) and directory trees.

XCOPY source [destination] [/A | /M] [/D:date] [/P] [/S] [/E] [/V] [/W]
  source       Specifies the file(s) to copy.
  destination  Specifies the location and/or name of new files.
  /A           Copies files with the archive attribute set,
               doesn't change the attribute.
  /M           Copies files with the archive attribute set,
               turns off the archive attribute.
  /D:date      Copies files changed on or after the specified date.
  /P           Prompts you before creating each destination file.
  /S           Copies directories and subdirectories except empty ones.
  /E           Copies any subdirectories, even if empty.
  /V           Verifies each new file.
  /W           Prompts you to press a key before copying.
  /Y           Suppresses prompting to confirm you want to overwrite an
               existing destination file.
  /-Y          Causes prompting to confirm you want to overwrite an
               existing destination file.

The switch /Y may be preset in the COPYCMD environment variable.
This may be overridden with /-Y on the command line

C:\>
```

You can use the FASTHELP command to see an abbreviated listing of a command's syntax and switches.

INSIDE DOS™

Inside DOS (ISSN 1049-5320) is published monthly by The Cobb Group.

Prices:

Domestic	$49/yr ($6.00 each)
Outside US	$69/yr ($8.50 each)

Phone:

Toll free	(800) 223-8720
Local	(502) 493-3300
Customer Relations Fax	(502) 491-8050
Editorial Department Fax	(502) 491-3433

Address: You may address tips, special requests, and other correspondence to

The Editor, *Inside DOS*
9420 Bunsen Parkway, Suite 300
Louisville, KY 40220

For subscriptions, fulfillment questions, and requests for bulk orders, address your letters to

Customer Relations
9420 Bunsen Parkway, Suite 300
Louisville, KY 40220

Copyright: Copyright © 1994, The Cobb Group. All rights reserved. *Inside DOS* is an independently produced publication of The Cobb Group. The Cobb Group reserves the right, with respect to submissions, to revise, republish, and authorize its readers to use the information submitted for both personal and commercial use.

The Cobb Group and its logo are registered trademarks of Ziff Communications Company. *Inside DOS* is a trademark of Ziff Communications Company. Microsoft and MS-DOS are registered trademarks and Microsoft Windows is a trademark of Microsoft Corporation. PC-DOS is a trademark of IBM Corporation.

Postmaster: Second class postage is pending in Louisville, KY. Send address changes to

Inside DOS
P.O. Box 35160
Louisville, KY 40232

Authorized Canada Post International Publications Mail (Canadian Distribution) Sales Agreement #XXXXXX CANADA GST #123669673. Send returns to Canadian Direct Mailing Sys. Ltd., 920 Mercer Street, Windsor, Ontario, N9A 7C2. Printed in the USA.

Staff:

Editor-in-Chief	Jim Stanfield
Contributing Editor	Van Wolverton
Editors	Linda Watkins Tiffany Tayor Tim Hampton
Production Artists	Tara Dickerson Tre Pryor
Circulation Manager	Tammy Castleman
Editorial Director	Jeff Yocom
Publishers	Mark Crane Jon Pyles

Advertising: For information about advertising in Cobb Group journals, contact Tracee Bell Troutt at (800) 223-8720, ext. 430.

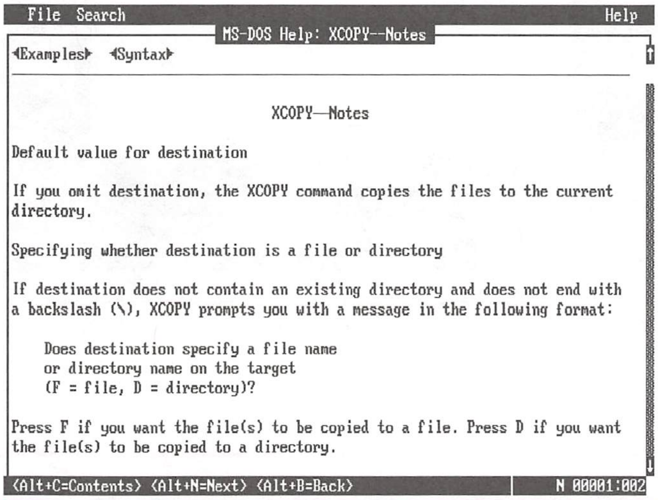
Back Issues: To order back issues, call Customer Relations at (800) 223-8720. Back issues cost $6.00 each, $8.50 outside the US. You can pay with MasterCard, VISA, Discover, or American Express, or we can bill you.

Bulk Sales: For information about bulk/group subscription sales, contact Larry Bolton at (800) 223-8720, ext. 387.

Advisory Board:
Earl Berry Jr.
Tina Covington
Marvin D. Livingood

upper-left corner of your screen, under the menu bar. When you do, DOS will display the screen shown in Figure C. Now, to see some examples of how to implement the XCOPY command, select Examples (or press E, then [Enter]) at the upper-left corner of your screen. DOS will then display the screen shown in Figure D. To return to the first screen (the one containing information on the syntax of the XCOPY command), simply select Syntax (or press S, then [Enter]) at the upper-left corner of your screen.

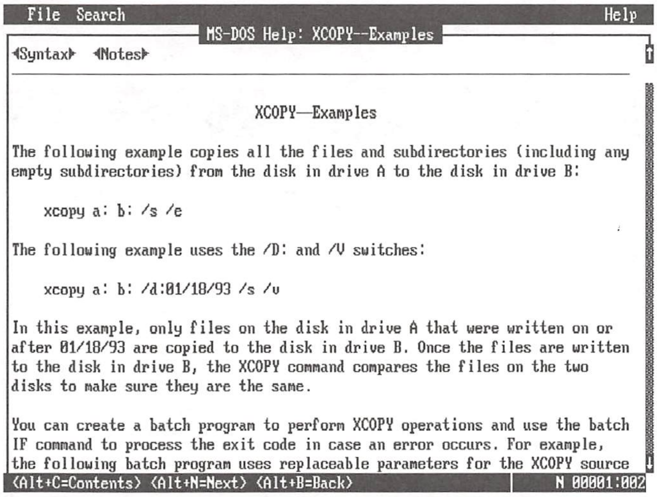
Figure C



```
 File  Search                                                      Help
                      MS-DOS Help: XCOPY--Notes
◄Examples►  ◄Syntax►

                          XCOPY—Notes

Default value for destination

If you omit destination, the XCOPY command copies the files to the current
directory.

Specifying whether destination is a file or directory

If destination does not contain an existing directory and does not end with
a backslash (\), XCOPY prompts you with a message in the following format:

    Does destination specify a file name
    or directory name on the target
    (F = file, D = directory)?

Press F if you want the file(s) to be copied to a file. Press D if you want
the file(s) to be copied to a directory.
<Alt+C=Contents> <Alt+N=Next> <Alt+B=Back>                     N 00001:002
```

DOS 6's Help utility offers more detailed information on a command's syntax.

Figure D



```
 File  Search                                                      Help
                    MS-DOS Help: XCOPY--Examples
◄Syntax►  ◄Notes►

                         XCOPY—Examples

The following example copies all the files and subdirectories (including any
empty subdirectories) from the disk in drive A to the disk in drive B:

    xcopy a: b: /s /e

The following example uses the /D: and /V switches:

    xcopy a: b: /d:01/18/93 /s /v

In this example, only files on the disk in drive A that were written on or
after 01/18/93 are copied to the disk in drive B. Once the files are written
to the disk in drive B, the XCOPY command compares the files on the two
disks to make sure they are the same.

You can create a batch program to perform XCOPY operations and use the batch
IF command to process the exit code in case an error occurs. For example,
the following batch program uses replaceable parameters for the XCOPY source
<Alt+C=Contents> <Alt+N=Next> <Alt+B=Back>                     N 00001:002
```

DOS 6's Help utility even gives examples of how to use a command.

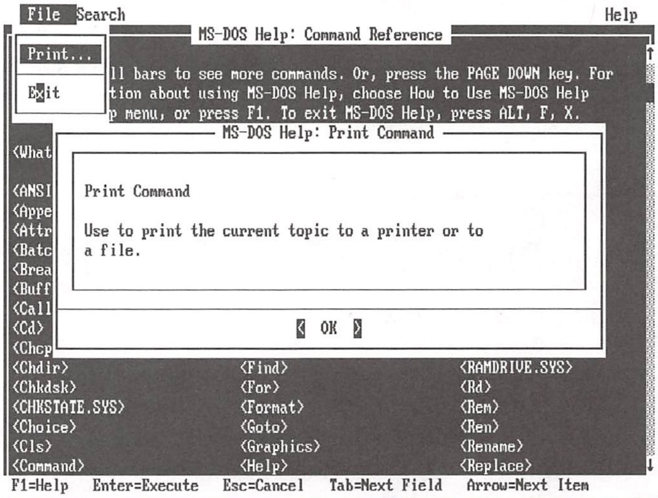
Exiting Help

Once you've read the Help screen, you can exit the Help utility by pulling down the File menu and selecting Exit. If you're not using a mouse, just press [Alt]fx. DOS will close the Help screen and return you to the command prompt.

Getting context-sensitive help

You can use the [F1] key to obtain context-sensitive help with most menu and dialog box options. For example, suppose you're working in the Help table of contents and you want to know what the Print... command (on the File menu) does. You can simply pull down the File menu and press [F1], since the Print... command is selected by default. DOS will display a brief explanation of the Print... command, as shown in Figure E. To dismiss the dialog box, click OK or press [Esc]. If you want to see a brief explanation of the Exit command, just press ⬇ once to highlight that command on the File menu, and press [F1]. DOS will display a dialog box containing an explanation of the Exit command.

Figure E



```
 File Search                                                       Help
                  MS-DOS Help: Command Reference
 Print...  ll bars to see more commands. Or, press the PAGE DOWN key. For
 Exit      tion about using MS-DOS Help, choose How to Use MS-DOS Help
           p menu, or press F1. To exit MS-DOS Help, press ALT, F, X.
                     MS-DOS Help: Print Command
<What
<ANSI   Print Command
<Appe
<Attr   Use to print the current topic to a printer or to
<Batc   a file.
<Brea
<Buff
<Call
<Cd>                          < OK >
<Chcp
<Chdir>                   <Find>                   <RAMDRIVE.SYS>
<Chkdsk>                  <For>                    <Rd>
<CHKSTATE.SYS>            <Format>                 <Rem>
<Choice>                  <Goto>                   <Ren>
<Cls>                     <Graphics>               <Rename>
<Command>                 <Help>                   <Replace>
F1=Help  Enter=Execute  Esc=Cancel  Tab=Next Field  Arrow=Next Item
```

DOS 6 offers context-sensitive help for most menus and dialog boxes.

New switches

As we mentioned, DOS 6's HELP command comes with four new switches. If you want, you can use the syntax

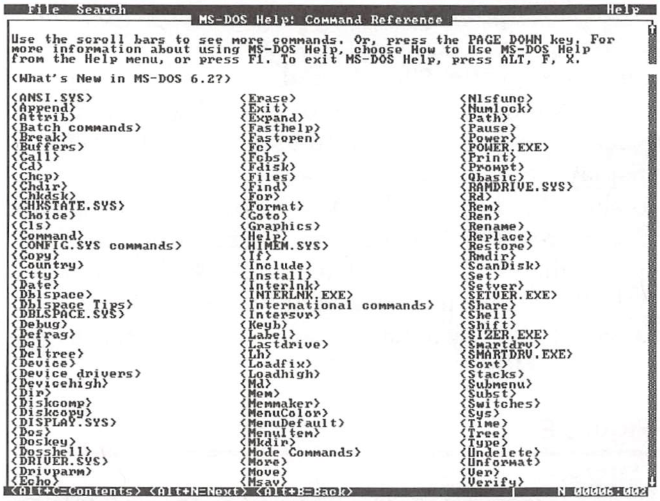
```
help command /b /g /h /nohi
```

You can use the /b switch to display the help text in black and white only. If you're using a CGA system, you can use the /g switch to improve the display of the help text. With the /h switch, you can switch from 25 lines per screen to 43 lines per screen for EGA systems and to 50 lines per screen for VGA or better systems. Figure F, on page 4, shows the Help table of contents in the 50-line-per-screen mode. If you're in the mood for color, you can use the /nohi switch, which instructs DOS to display the help text using eight low-intensity colors.

Searching the help text

You can search the help text for a string by selecting the Find... command from the Search menu. When you select this command, DOS will display the Find dialog box shown in Figure G. You enter the text you want to find in the Find What text box. (You can enter up to 127 characters in this text box.) Then, you can tell DOS to make the search case-sensitive by checking

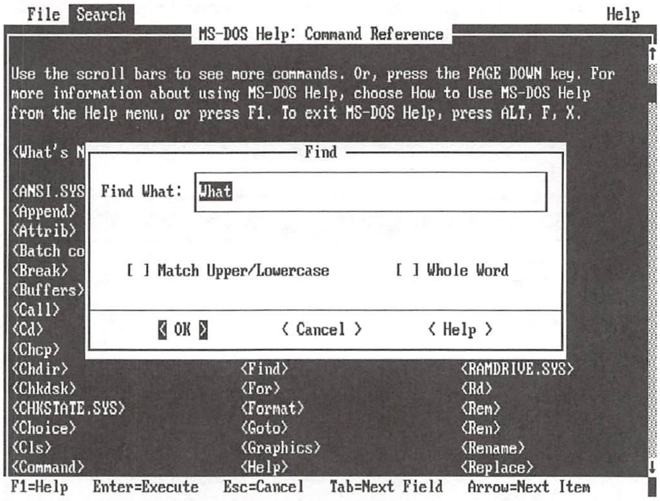
Figure F



```
  File  Search                                                         Help
                        MS-DOS Help: Command Reference
Use the scroll bars to see more commands. Or, press the PAGE DOWN key. For
more information about using MS-DOS Help, choose How to Use MS-DOS Help
from the Help menu, or press F1. To exit MS-DOS Help, press ALT, F, X.

<What's New in MS-DOS 6.2?>

<ANSI.SYS>                  <Erase>                    <Nlsfunc>
<Append>                    <Exit>                     <Numlock>
<Attrib>                    <Expand>                   <Path>
<Batch commands>            <Fasthelp>                 <Pause>
<Break>                     <Fastopen>                 <Power>
<Buffers>                   <Fc>                       <POWER.EXE>
<Call>                      <Fcbs>                     <Print>
<Cd>                        <Fdisk>                    <Prompt>
<Chcp>                      <Files>                    <Qbasic>
<Chdir>                     <Find>                     <RAMDRIVE.SYS>
<Chkdsk>                    <For>                      <Rd>
<CHKSTATE.SYS>              <Format>                   <Rem>
<Choice>                    <Goto>                     <Ren>
<Cls>                       <Graphics>                 <Rename>
<Command>                   <Help>                     <Replace>
<CONFIG.SYS commands>       <HIMEM.SYS>                <Restore>
<Copy>                      <If>                       <Rmdir>
<Country>                   <Include>                  <ScanDisk>
<Ctty>                      <Install>                  <Set>
<Date>                      <Interlnk>                 <Setver>
<Dblspace>                  <INTERLNK.EXE>             <SETVER.EXE>
<Dblspace Tips>             <International commands>   <Share>
<DBLSPACE.SYS>              <Intersvr>                 <Shell>
<Debug>                     <Keyb>                     <Shift>
<Defrag>                    <Label>                    <SIZER.EXE>
<Del>                       <Lastdrive>                <Smartdrv>
<Deltree>                   <Lh>                       <SMARTDRV.EXE>
<Device>                    <Loadfix>                  <Sort>
<Device drivers>            <Loadhigh>                 <Stacks>
<Devicehigh>                <Md>                       <Submenu>
<Dir>                       <Mem>                      <Subst>
<Diskcomp>                  <Memmaker>                 <Switches>
<Diskcopy>                  <MenuColor>                <Sys>
<DISPLAY.SYS>               <MenuDefault>              <Time>
<Dos>                       <MenuItem>                 <Tree>
<Doskey>                    <Mkdir>                    <Type>
<Dosshell>                  <Mode Commands>            <Undelete>
<DRIVER.SYS>                <More>                     <Unformat>
<Drivparm>                  <Move>                     <Ver>
<Echo>                      <Msav>                     <Verify>
<Alt+C=Contents> <Alt+N=Next> <Alt+B=Back>                     N 00006:002
```

You can use the HELP command's /h switch to change from the default 25 lines per screen to 50 lines per screen.

Figure G



```
 File  Search                                                          Help
                        MS-DOS Help: Command Reference
Use the scroll bars to see more commands. Or, press the PAGE DOWN key. For
more information about using MS-DOS Help, choose How to Use MS-DOS Help
from the Help menu, or press F1. To exit MS-DOS Help, press ALT, F, X.

<What's N                        Find
<ANSI.SYS   Find What:  [What                                   ]
<Append>
<Attrib>
<Batch co
<Break>        [ ] Match Upper/Lowercase           [ ] Whole Word
<Buffers>
<Call>
<Cd>                  < OK >           < Cancel >         < Help >
<Chcp>
<Chdir>                     <Find>                     <RAMDRIVE.SYS>
<Chkdsk>                    <For>                      <Rd>
<CHKSTATE.SYS>              <Format>                   <Rem>
<Choice>                    <Goto>                     <Ren>
<Cls>                       <Graphics>                 <Rename>
<Command>                   <Help>                     <Replace>
F1=Help  Enter=Execute  Esc=Cancel  Tab=Next Field  Arrow=Next Item
```

You can use the Find dialog box to search for a character or string within the Help file.

Getting help when you're not sure what you need

In the accompanying article, we showed you how to get help when you know the command you want help with. However, if you're unsure of the exact command you need, you can see a table of contents listing all the commands and device drivers that have help text available. To view this table of contents, simply type

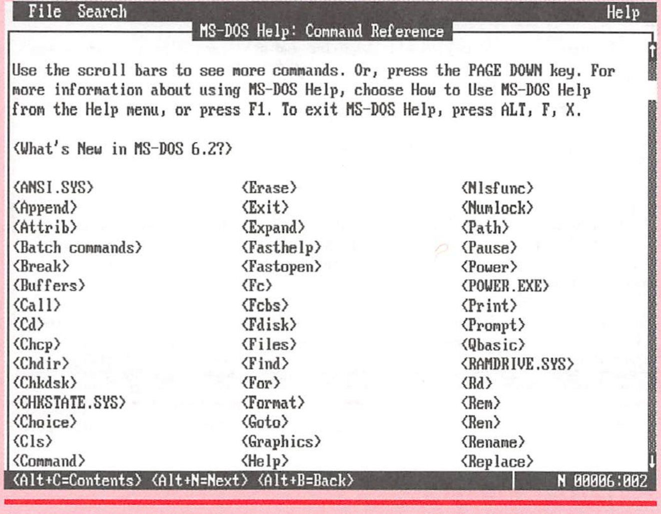
```
c:\>help
```

at the command prompt. When you do, DOS will display the screen shown in Figure A. To view more commands, you can use the scroll bars at the right side of the window, or you can press the [Page Down] key.

To see the help text associated with a command, simply click on the command name. If you don't have a mouse, you can use the arrow keys or press [Tab] to move your cursor to a command name, and then press [Enter]. In addition, you can press a letter to make DOS position the cursor under the first command that begins with that letter.

After you've read the help text, you can return to the table of contents by pressing [Esc] or by pressing [Alt]c. From there, you can view the help text for another command or quit the Help utility. To quit the Help utility and return to the command prompt, simply pull down the File menu and select Exit or press [Alt]fx.

Figure A



```
 File  Search                                                          Help
                        MS-DOS Help: Command Reference
Use the scroll bars to see more commands. Or, press the PAGE DOWN key. For
more information about using MS-DOS Help, choose How to Use MS-DOS Help
from the Help menu, or press F1. To exit MS-DOS Help, press ALT, F, X.

<What's New in MS-DOS 6.2?>

<ANSI.SYS>                  <Erase>                    <Nlsfunc>
<Append>                    <Exit>                     <Numlock>
<Attrib>                    <Expand>                   <Path>
<Batch commands>            <Fasthelp>                 <Pause>
<Break>                     <Fastopen>                 <Power>
<Buffers>                   <Fc>                       <POWER.EXE>
<Call>                      <Fcbs>                     <Print>
<Cd>                        <Fdisk>                    <Prompt>
<Chcp>                      <Files>                    <Qbasic>
<Chdir>                     <Find>                     <RAMDRIVE.SYS>
<Chkdsk>                    <For>                      <Rd>
<CHKSTATE.SYS>              <Format>                   <Rem>
<Choice>                    <Goto>                     <Ren>
<Cls>                       <Graphics>                 <Rename>
<Command>                   <Help>                     <Replace>
<Alt+C=Contents> <Alt+N=Next> <Alt+B=Back>                     N 00006:002
```

Type help *at the command prompt to view this Help's table of contents.*

the Match Upper/Lowercase option. Case-sensitive searches will most often be faster than case-insensitive ones. An example of a case-sensitive search is telling DOS to find the string *Cls* instead of *CLS* or *cls*.

If you want, you can check the Whole Word option to instruct DOS to ignore any matches that are within a word or string. (Whole-word searches recognize a match only when the text is surrounded by spaces, punctuation marks, or other special characters, like parentheses.) So, a whole-word search that's supposed to find the string *ansi* will ignore the strings *ansi.sys* and *<ansi>*.

When you invoke a search, DOS begins at the current cursor position and moves forward through the Help file. If DOS comes to the end of the file without finding a match, it will move to the beginning of the file and continue its search. When DOS finds a match, it will place the cursor under the first letter of the matching string. ❖

LAPTOP COMPUTING

Putting POWER at your fingertips

VERSION 6.x

By Robert L. Hummel

If your laptop is running short of power, don't panic. One of DOS 6's hidden utilities may hold the solution for you. My laptop is a dinosaur. It's big, it's old, and it's heavy. The battery pack alone weighs over five pounds—more than the total weight of most modern laptops. But even with that massive battery, I can't operate my laptop for more than about an hour and a half without reaching for the charger. The immense power requirements of its 80386DX CPU, 80387 math coprocessor, backlit screen, hard drive, and cooling fan ensure that I'll never take the power of portable computing too far from civilization.

Over the past few years, the situation has changed dramatically. Manufacturers such as Intel and AMD have begun to make low-power CPUs that use only a fraction of the power that their desktop cousins gulp. But power-saving CPUs remain just one aspect of the total energy consumption that occurs in your computer. True power management has to be implemented on a systemwide basis to have a maximum effect.

If you've upgraded to DOS 6.x, you can use a little-known utility that might help you extend the battery life of your laptop and even help your desktop PC run cooler and use less power. The POWER.EXE device driver, which comes with DOS 6, can reduce your PC's power consumption when your applications and devices are idle. For POWER.EXE to do its best work, your PC's BIOS must conform to the Advanced Power Management (APM) specification. To see why, let's briefly review the concept of power management.

Power management

The idea of power conservation isn't new. Although my old laptop is power mad, it came with a device driver that turns off the display when I don't press a key for some set time. It also saves power by turning off the hard drive motor in the absence of disk activity. These types of power-conservation functions are relatively easy to implement because any external activity, such as pressing a key, gives the PC a chance to reactivate the system. A crucial requirement of successful power management is that it be transparent to applications and to the user.

Intel incorporated sophisticated system and power management features into special versions of its CPUs. Specifically, Intel designed the 386SL and 486SL processors to integrate and coordinate systemwide power management by using a system management mode (SMM). Because power management is built in at the system level, PCs using the Intel chips are able to control power use transparently—without requiring any modifications to the operating system or applications software.

A CPU with an SMM can provide some amazing capabilities. During idle periods, you can slow or even stop the CPU and later restart it without losing any data. You can shut down and reactivate individual peripherals, such as hard disks, displays, modems, and even serial and parallel ports, as needed. Because the SMM functions are a superset of the normal CPU's functions, they are compatible with even the most ill-behaved programs.

The APM specification

Hardware-based power management has proven effective in reducing the power that laptops and PCs consume. But a hardware-only solution suffers from a crucial deficiency: Hardware operates in complete ignorance of your software; it can't tell what the software is doing or—more important—what it's going to do.

Let's say that your database saves a file to disk. The SMM intercepts the request and brings the disk up to speed; then, the application proceeds normally and transfers the file. Assume that the entire process takes about 10 seconds. The database is now finished with the disk, but the hardware activity timer won't expire for another 2 minutes and 50 seconds. The result is a lot of wasted power over a typical computing session. In addition, the delay you incur while waiting for the hard drive to come up to speed can be disconcerting.

Now imagine that your database has the ability to tell the system two important things: It can prompt the system to ready the hard disk for use, and it can notify the system that it's finished with the hard disk immediately after saving the file. The system can then bring the disk up to speed when it's needed and shut it down immediately when it's not. This ability not only saves a great deal of precious power but also makes your power-saving laptop seem peppier.

Intel and Microsoft introduced the APM specification in early 1992. This specification establishes a standard way for software to communicate with hardware and control hardware's use of power. Software written to take advantage of the APM interface can help your computer increase its efficiency. APM doesn't replace traditional power-management techniques. Instead, it enhances them by enabling the operating system and applications to share power-management responsibility with the hardware.

Fully implementing APM requires BIOS and operating system support. Fortunately, the APM specification is nonproprietary, and several BIOS vendors already incorporate APM code in their products. Both DOS 6 and recent versions of Windows offer operating support. According to Microsoft, hardware conforming to APM can produce power savings of up to 25 percent, and nonconforming hardware can still realize savings of up to 5 percent.

Using POWER.EXE

The POWER.EXE device driver, provided with DOS 6, activates Microsoft's APM system. Although it has an EXE extension, POWER.EXE is really a device driver with a small executable program built in. Assuming that you keep your DOS files in the C:\DOS subdirectory, you can enable APM by editing your CONFIG.SYS file to include the following line:

```
DEVICE=C:\DOS\POWER.EXE
```

The next time you boot your system, the device driver POWER.EXE will load, activating the APM power conservation interface. There's no need to specify the DEVICEHIGH statement—the driver will automatically load into upper memory if available. If you want to force the driver into low memory, add the /LOW switch to the command line.

Once you've loaded the POWER.EXE driver, you can adjust the current power settings by executing POWER as a normal DOS command from the command prompt or in a batch program. The full syntax of the POWER command is

```
POWER [ ADV[:MAX¦REG¦MIN] ¦ STD ¦ OFF]
```

Let's examine these options by working backwards. First, the function of the OFF option should be clear: It turns off all power management. You might want to choose this option to elicit maximum response from your PC in situations where power conservation isn't an issue.

The STD (standard) option disables the advanced management functions POWER provides but leaves the hardware power-management features of your PC enabled. If, however, your PC doesn't support the APM specification, the STD option has the same effect as the OFF option.

The ADV (advanced) option enables aggressive power conservation. This option's three possible settings (MAX, REG, and MIN) accomplish this in varying degrees. By default, the POWER.EXE driver loads with the ADV:REG option enabled. To enable maximum power conservation, use the ADV:MAX option. In some cases, the MAX and REG settings will make your applications or hardware respond poorly. If so, use the ADV:MIN setting. Once you determine the most effective setting for your computer and your computing habits, you can make it the default by adding it to the DEVICE command line in your CONFIG.SYS file.

POWER has an interesting benefit for the curious. If you've loaded the POWER.EXE device driver, typing *POWER* at the command prompt with no arguments will produce a display similar to

```
Power Management Status
---------------------------
Setting = ADV: MAX
CPU: idle 69% of time.
```

Here, the program indicates that it's using maximum power compression. It also reports a utilization figure for the CPU. Unfortunately, the algorithm the program uses to calculate this figure is undocumented, but it appears to be the ratio of the amount of time the CPU has been idle to the amount of time the POWER.EXE driver has been active.

The future

APM power conservation techniques are extending the working life of laptop batteries. Manufacturers are also applying them to desktop systems not only to reduce power use but also to increase component life by reducing the wear on moving components and limiting the damage that excessive heat causes. ❖

Robert Hummel is president of Schaefer Software, a New Hampshire-based consulting firm. He's the author of several programming books and a regular contributor to The DOS Authority, *another Cobb Group journal.*

BATCH FILE

Blanking your screen

VERSION
5.0 & 6.x

If you want to keep passers-by from casually observing what you've been working on, you can use a utility to blank your screen. If you don't have such a utility handy, you can create a batch file to do the trick.

In this article, we'll show you how to create two batch files that blank your screen when you need to leave your PC. (You should store these batch files in a directory you've included in the PATH statement in your AUTOEXEC.BAT file.) Although you can use the CLS command to remove all traces of your work from your screen, you'll find that the batch files in this article will clear your screen a little more elegantly.

BLANK.BAT

The first batch file we'll create blanks your screen and redisplays your work when you press a key. To create the batch file, you first need to open the DOS Editor or any other word processing program that's capable of creating an ASCII text file without adding any formatting commands. Then, type the text shown in Figure A, save the file as BLANK.BAT, and exit the program.

The batch file starts with the @ECHO OFF command, which prevents the commands from displaying as the batch file runs. Next, two REM statements give the name and purpose of the batch file. The CLS command on the fourth line clears your screen.

Figure A

```
@echo off
rem BLANK.BAT blanks your screen until
rem you press a key.
cls
pause >nul
```

This batch file blanks your screen until you press a key.

The next command

```
pause >nul
```

performs two actions: It halts the execution of the macro until you press a key; then, it displays the message *Press any key to continue . . .*, followed by a blinking cursor under the message. So, phase one of our batch file—blanking your screen until you press a key—works fine. However, the message's display defeats the purpose of blanking the screen. Therefore, we redirected the output of the PAUSE command to the NUL device by using >nul. (DOS defines NUL as a nonfunctioning device.) With the message gone, all that's left onscreen is a blinking cursor. As a result, you've effectively blanked your screen.

To run this batch file, simply type *blank* at the command prompt and press [Enter]. The batch file will clear your screen and display a blinking cursor in the upper-left corner of your screen. To redisplay your command prompt, just press any key.

Better blanking and added security to boot

The OFF.BAT file, shown in Figure B, not only blanks your screen but also discourages another user from entering commands at the DOS command prompt. OFF.BAT simply changes the color of the text on your screen from white to black. Therefore, as long as you're using the default screen display—white text on a black background—your screen will display black letters on a black background when you run OFF.BAT. This blanks your screen and discourages others from using your PC, since they can't see any letters they type at the command prompt. (Of course, you'll need to reset your screen's display once you return to your PC; we'll show you how to do this later by using the file ON.BAT.)

Figure B

```
@echo off
rem OFF.BAT blanks your screen by
rem displaying black letters on a black background.
echo ←[30m
cls
```

You can use OFF.BAT to blank your screen.

To change the colors your screen displays, you must have ANSI.SYS running. (ANSI.SYS is a device driver you can use to customize your keyboard and to control the colors on your screen and the position of your cursor.) To have ANSI.SYS running every time you use your computer, you can place the command

```
device=\dos\ansi.sys
```

or

```
devicehigh=\dos\ansi.sys
```

in your CONFIG.SYS file.

The first three lines of OFF.BAT prevent the commands from displaying as the batch file runs, and they show the name and purpose of the batch file. The command

```
echo ←[30m
```

changes the color of your display's foreground to black. (The ANSI code for a black foreground color is 30.) By the way, to enter the ← character, simply press [Ctrl]p and then press [Esc].

If you've customized your screen to display a color scheme other than white text on a black background, you can still use our OFF.BAT file. Simply replace 30 in the ECHO command with the ANSI code that represents the color of your screen's background.

The last command, CLS, displays a fresh screen with the command prompt at the upper-left corner. However, since the prompt will appear in black, your screen will appear blank. At this point, you won't be able to see anything you type at the command prompt.

If you run a program that has its own color scheme, such as Edit, the program will display in its default colors. However, once you exit the program, your screen will again appear blank.

To reset your screen's display, create ON.BAT, shown in Figure C. This file returns your screen's foreground color to white.

Figure C

```
@echo off
rem ON.BAT unblanks your screen by
rem resetting your screen's color scheme.
echo ←[0m
cls
```

You use ON.BAT to "unblank" your screen.

As with OFF.BAT, the first three lines of ON.BAT turn off the display of text onscreen and show the name and purpose of the batch file. Then, the command

```
echo ←[0m
```

resets your screen's color scheme to the default white text on a black background. If you're using a different color scheme, replace 0 with the ANSI code for your custom color scheme. Finally, the CLS command clears any text that was entered after you executed OFF.BAT and places your command prompt at the upper-left corner of your screen. ❖

Got a boot problem?

If DOS won't fully boot your PC and displays a message telling you that you have a bad or missing command interpreter, then boot your computer with an emergency boot-up floppy disk. If you don't have one, go to a neighboring computer, insert a floppy disk into drive a: and type *format a:/s*. Then, insert this disk into your PC and reboot. (The /s switch tells DOS to make a system disk containing COMMAND.COM and some important DOS commands.)

VAN WOLVERTON

Saving space on your hard disk

VERSION
5.0 & 6.x

The steep decline in the cost of hard disks over the past few years has been a great boon to increased speed and convenience. Anyone who's been around long enough to remember running a system with just two floppy disks would willingly suffer a three- or four-time reduction in processor speed if their only other choice was to give up their hard disk.

A decade ago, I paid more than $800 for a 20-Mb hard disk that completely filled a 5¼" drive bay and wasn't a whole lot faster than a floppy disk—average access time was something over 80 milliseconds. This past summer, I bought a 546-Mb hard disk that wasn't much bigger than a deck of cards and had an average access time of 12 milliseconds. For this wondrous device, I paid $277. Amazing.

All this progress isn't without a few problems—however easy they are to overlook. Keeping track of files can be a challenge when a disk contains several thousand of them, so it's really important to set up and maintain a good file structure. More ominously, a hard disk failure—always a painful experience—becomes even more ruinous as disk capacity increases. The imperative to back up regularly has never been greater.

Allocating space on a hard disk

Another issue related to the size of your hard disk—more accurately, the size of your hard disk partition—is how efficiently DOS uses hard disk space. Although the output of the DIR command shows you the size of a file as an exact number of bytes, DOS parcels out space to files in fixed chunks, called *allocation units* or, sometimes, *clusters*. The size of the allocation unit depends on the size of your disk partition, as shown in Figure A.

The size of an allocation unit is the minimum amount of space a file takes up on a hard disk, even if it contains only 1 byte of data. As Figure A shows, every file on my 20-Mb hard disk occupied at least 2 Kb, but that minimum jumps to 16 Kb on my new 546-Mb drive.

Each file will waste about half an allocation unit. If I stored 500 files on that old 20-Mb drive, 500 Kb would be allocated to files but not used; that seems like an acceptable price for speed and convenience. But the price goes up rapidly with increased partition size. I've got 7,500 files stored on my new hard disk, and its allocation unit is 16 Kb, so each file wastes 8 Kb. That means I'm losing the use of about 60 Mb—that's right, Mb—three times the total capacity of that older hard disk! The price of speed and convenience has escalated right along with the price of the hard disks themselves.

Figure A

Partition (Megabytes)	Allocation Unit (Bytes)
less than 128	2,048 (2 Kb)
128 to 255	4,096 (4 Kb)
256 to 511	8,192 (8 Kb)
512 or greater	16,384 (16 Kb)

The minimum space DOS allocates for a file depends on the size of the disk partition.

Cut waste with smaller partitions

You don't have to accept these losses. DOS lets you partition your hard disk however you like, using the FDISK command. So, you can reduce the size of the allocation unit by partitioning a large hard disk into two or more smaller volumes. If I partitioned my 546-Mb hard disk into two 273-Mb drives, for example, I would cut the losses in half because the allocation unit would change from 16 Kb to 8 Kb. If I partitioned the disk into three 182-Mb drives, the allocation unit would drop to 4 Kb, cutting the waste by three-fourths.

But changing the partition size means losing the files on my hard disk, and I've got 7,500 files in more than 400 directories. I really don't want to back up all those files, then repartition and reformat the disk. And besides, I've still got 120 Mb available, so the lost 60 Mb right now is a problem more in principle than in practice.

But just because I'm lazy doesn't mean I'm stuck. I can gain space on my hard disk by backing up the files, using FDISK to define smaller partitions, and restoring the files.

Cut waste with fewer files

Whether or not you reduce your partition size, you can reduce the amount of unused hard disk space by cutting the number of files stored on the disk. There are several ways to do this:

- Delete all files you don't need.
- Use a file-compression utility, such as PKZIP, to compress and combine several files into one.
- Use batch file techniques that reduce the number of files needed.

Unneeded files invariably proliferate, especially on large hard disks. In addition to the obvious ones, such as BAK files and old versions of data files, some are easy to forget—such as programs you no longer use, temporary files, or even directories you created and meant to erase right away. It's time consuming, but a trip through all your directories can often yield tens or hundreds of files you can delete. You could probably double that number by copying infrequently used files to a floppy disk and deleting them from your hard disk.

File compression is a real space saver

File-compression programs reduce the size of a file by using the same techniques as disk compression programs. File-compression programs also let you combine more than one file into a single compressed file, so you save space not only by making the files smaller but also by reducing the number of files on your hard disk. (If you're using a disk-compression utility such as Stacker or the ill-fated DoubleSpace, then a file-compression program can't save you any space, so you can skip this topic.)

For example, I archive all the columns I write for *Inside DOS* in COLUMNS.ZIP, a compressed file. Currently, that file contains 51 compressed files in 278,581 bytes, which is 17.003234 allocation units. The actual amount of space the file takes up on the hard disk, therefore, is 18 allocation units, or 294,912 bytes. The 51 uncompressed files would take up 59 allocation units (43 files are less than 1 allocation unit long, and 8 are between 1 and 2 allocation units). Compressing the files, therefore, is saving me 41 allocation units (59 less 18), or 671,744 bytes.

PKZIP comes in both shareware and commercial versions. Look in a software store or mail-order house for the commercial version. For the shareware version, check with a user's group or bulletin board system (including national services such as CompuServe or America Online).

Batch-file techniques that save space

Finally, if you use batch files quite a bit, you can save disk space by following a couple of guidelines:

- If a batch file displays instructions or a menu, use ECHO commands instead of displaying a separate file with the TYPE command.

- Combine several functions in one batch file instead of using several small batch files.

If a batch file displays a long message, it's usually easier to put the message in a separate file and display it with a TYPE command than to use several ECHO commands. On a large hard disk, however, each extra file costs you almost 8 Kb or 16 Kb of disk space (depending on whether the disk is larger or smaller than 512 Mb), regardless of the length of the message. The choice between the ECHO and TYPE command techniques depends on how crowded your disk is and how often you display long messages.

Avoid a proliferation of small batch files

Some uses of batch files involve many small batch files. Early menu systems, for example, used one batch file to display several choices, then a separate batch file for each choice. For example, a batch file might clear the screen, display the following message, and return to the DOS prompt:

```
MAIN MENU

1. Lotus 123
2. Quicken
3. WordPerfect

Type 1, 2, or 3.
```

DOS is simply waiting for you to type a command. If you type *1*, *2*, or *3*, the corresponding menu choice is carried out by batch files named 1.BAT, 2.BAT, and 3.BAT. Occasionally, computer stores still deliver systems with such a menu system installed on the hard disk, sometimes with half a dozen or more choices. This has the potential to waste quite a bit of disk space.

If you have a menu system like this and you're using DOS 6, you can eliminate the need for the extra files by using the CHOICE command to get the response and the IF ERRORLEVEL and GOTO commands to carry out the menu choices.

Housekeeping is your responsibility

If you're using version 6 (and it's worth the upgrade), don't forget to run ScanDisk and Defrag once a week or so to check for any hard disk problems that might be looming and to keep the files as compact as possible. Defragmenting lets you improve disk performance and use storage efficiently.

Conclusion

It's easy to let today's big, fast hard disks lull us into some sloppy housekeeping that can fill our disks much more quickly than we might think possible. By using the techniques described here, you can put off the day when your hard disk fills up. ❖

Larger characters at the command prompt make reading easier

Although I hate to admit it, my age is catching up with me. It seems the older I get, the worse my vision becomes. Is there any way to make the characters at the command prompt larger? Right now, in order to read my screen, I have to get a little more intimate with it than I'd like.

Paula Horner
Cincinnati, Ohio

Fortunately, there's a way to make the letters on your screen larger when you're working at the command prompt. You can use DOS's MODE command, which lets you configure a variety of devices, one of which is your screen.

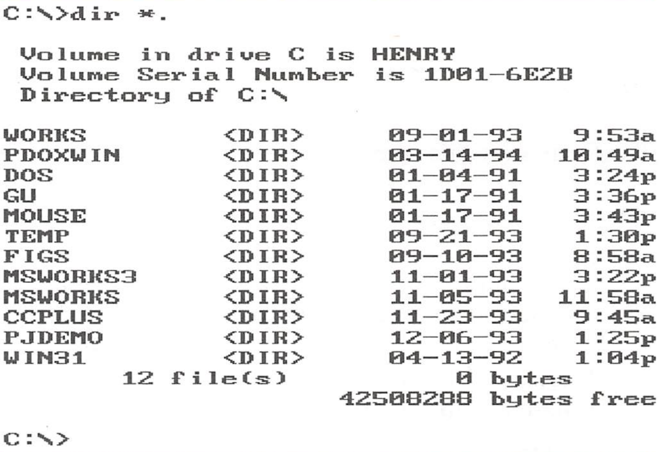
To make the letters larger, you simply enter the command

```
c:\>mode 40
```

at the command prompt. When you do, DOS will change the number of columns you can display on your screen from 80 to 40. Consequently, the size of the letters on your screen will increase to fit this display mode, as shown in Figure A.

This MODE command will affect the size of the characters on your screen only while you're working at the command prompt. For example, if you change to the 40-column mode and then open DOS Edit, the screen will revert to the 80-column mode. When you close Edit, your screen will again appear in the 40-column mode.

Figure A



```
C:\>dir *.

 Volume in drive C is HENRY
 Volume Serial Number is 1D01-6E2B
 Directory of C:\

WORKS        <DIR>      09-01-93   9:53a
PDOXWIN      <DIR>      03-14-94  10:49a
DOS          <DIR>      01-04-91   3:24p
GU           <DIR>      01-17-91   3:36p
MOUSE        <DIR>      01-17-91   3:43p
TEMP         <DIR>      09-21-93   1:30p
FIGS         <DIR>      09-10-93   8:58a
MSWORKS3     <DIR>      11-01-93   3:22p
MSWORKS      <DIR>      11-05-93  11:58a
CCPLUS       <DIR>      11-23-93   9:45a
PJDEMO       <DIR>      12-06-93   1:25p
WIN31        <DIR>      04-13-92   1:04p
       12 file(s)              0 bytes
                        42508288 bytes free

C:\>
```

You can increase the size of the characters at the command prompt with the MODE command.

To reset your screen to the default 80-column mode, simply enter the command

```
c:\>mode 80
```

at the command prompt.

Squeezing out a little more memory

Recently, I installed a game on my PC, but I can't run it because DOS 6 tells me there's not enough memory. I used MemMaker to optimize my system's memory configuration, but doing this merely enabled me to run the game very slowly. Is there anything I can do to gain a little more memory and improve the game's performance?

Randy Napier
Tucson, Arizona

There are a few more things you can try in order to squeeze out a little more memory. First, in your AUTOEXEC.BAT and CONFIG.SYS files, remark out any device drivers or commands you don't need for the game, such as SHARE.EXE (if you're not running the game under Windows), ANSI.SYS, DOSKEY, and SETVER.EXE.

Second, you can reduce the amount of memory DOS reserves for open files. Open your CONFIG.SYS file to see how many file handles are specified. Look for the line that contains the command FILES along with an equal to sign. If the number after = is 20 or less, you probably won't want to reduce the number of file handles. However, if the number is greater than 20, you can reduce the value in the FILES= command and reboot your computer. If DOS displays a message saying there aren't enough file handles available, then increase the number until your PC boots properly.

Third, you can try reducing the amount of memory DOS sets aside for disk buffers. Each disk buffer takes up about 532 bytes. So, each time you reduce the first value in your BUFFERS command by 1, you gain about 532 bytes of memory.

You may also want to try reducing the number of stacks DOS reserves for handling hardware interrupts.

The default STACKS setting in CONFIG.SYS is

```
STACKS=9,128
```

The format for this command is

```
STACKS=number of stacks,bytes per stack
```

If you're running Windows, however, you'll need a STACKS setting of at least 9, with 256 bytes per stack.

Now, let's modify a few of the settings in the AUTOEXEC.BAT and CONFIG.SYS files and see how those changes can save memory. Figure A shows our original AUTOEXEC.BAT and CONFIG.SYS files, along with information on the memory usage resulting from the configurations in these two files. Figure B shows our modified AUTOEXEC.BAT and CONFIG.SYS files, along with information on the memory usage resulting from the new configurations. As you can see by comparing the total free memory in Figures A and B, we gained 24 Kb of memory!

Before you modify your AUTOEXEC.BAT and CONFIG.SYS files, you should save their original contents by copying each file to a temporary file, such as AUTOEX2.BAT and CONFIG2.SYS, respectively. That way, you can restore your original settings by copying the contents of AUTOEX2.BAT and CONFIG2.SYS back into AUTOEXEC.BAT and CONFIG.SYS. ❖

Figure A

```
C:\type autoexec.bat
LH /L:0;1,42416 /S C:\DOS\SMARTDRV.EXE
C:\DOS\SHARE.EXE /l:500 /f:5100
@ECHO OFF
PROMPT $P$G
PATH
C:\DOS;C:\;C:\UTILS;C:\WIN31;C:\WIN31\SYSTEM;C:\GU;C:\GU\COLLAGE;C:\WORKS
PATH C:\DOS;%PATH%;C:\ODAPI
SET TEMP=C:\WIN31\TEMP
LH /L:1,12928 C:\DOS\FASTOPEN C:=130
LH /L:1,56928 C:\DOS\mouse.COM /Y
cd\

C:\type config.sys
DEVICE=C:\DOS\SETVER.EXE
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE NOEMS 512
FILES=40
DOS=UMB
LASTDRIVE=E
FCBS=4,0
REM DEVICEHIGH C:\DOS\SMARTDRV.SYS 1024
DEVICEHIGH C:\DOS\ANSI.SYS
DOS=HIGH
STACKS=12,256
SHELL=C:\DOS\COMMAND.COM C:\DOS\ /E:512 /p
BUFFERS=15,0

C:\mem
Memory Type         Total  =   Used  +   Free
----------------  -------   -------   -------
Conventional         640K       53K      587K
Upper                155K       46K      109K
Reserved               0K        0K        0K
Extended (XMS)     3,301K    1,241K    2,060K
----------------  -------   -------   -------
Total memory       4,096K    1,340K    2,756K

Total under 1 MB     795K       99K      696K

Largest executable program size      587K (600,896 bytes)
Largest free upper memory block      109K (111,200 bytes)
MS-DOS is resident in the high memory area.
```

The memory usage shown above is the result of our original AUTOEXEC.BAT and CONFIG.SYS files' configuration.

Figure B

```
C:\type autoexec.bat
LH /L:0;1,42416 /S C:\DOS\SMARTDRV.EXE
rem C:\DOS\SHARE.EXE /l:500 /f:5100
@ECHO OFF
PROMPT $P$G
PATH
C:\DOS;C:\;C:\UTILS;C:\WIN31;C:\WIN31\SYSTEM;C:\GU;C:\GU\COLLAGE;C:\WORKS
PATH C:\DOS;%PATH%;C:\ODAPI
SET TEMP=C:\WIN31\TEMP
LH /L:1,12928 C:\DOS\FASTOPEN C:=130
LH /L:1,56928 C:\DOS\mouse.COM /Y
cd\

C:\type config.sys
rem DEVICE=C:\DOS\SETVER.EXE
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE NOEMS 512
FILES=20
DOS=UMB
LASTDRIVE=E
FCBS=4,0
REM DEVICEHIGH C:\DOS\SMARTDRV.SYS 1024
rem DEVICEHIGH C:\DOS\ANSI.SYS
DOS=HIGH
STACKS=9,256
SHELL=C:\DOS\COMMAND.COM C:\DOS\ /E:512 /p
BUFFERS=10,0

C:\mem
Memory Type         Total  =   Used  +   Free
----------------  -------   -------   -------
Conventional         640K       33K      607K
Upper                155K       42K      113K
Reserved               0K        0K        0K
Extended (XMS)     3,301K    1,241K    2,060K
----------------  -------   -------   -------
Total memory       4,096K    1,316K    2,780K

Total under 1 MB     795K       75K      720K

Largest executable program size      607K (621,568 bytes)
Largest free upper memory block      113K (115,424 bytes)
MS-DOS is resident in the high memory area.
```

By modifying our AUTOEXEC.BAT and CONFIG.SYS files, we gained memory.

Printed in the USA
This journal is printed on recycled and recyclable paper.
50% recycled fiber including 10% post-consumer waste paper